IMT574 Problem Set 5

Your name:

Deadline: Mar 7 midnight

Introduction

This problem set should be easier than the gradient descent one, but it is still a lot of the easier stuff.... You are mostly asked just to use existing packages to analyze a big survey dataset. You should also compute accuracy, precision, recall, and F-score, and make a ROC curve. These tasks you will implement yourself without any existing libraries!

The problem set has two parts: first you try to get as good a prediction as you can using k-NN, logistic regression and SVM methods, and thereafter you analyze how much does knowledge of country of the respondent improve our prediction.

The problem set has 4 goals: a) be better working with pandas and data; b) explore classification methods we have learned; c) learn about classification goodness measures; and d) learn a little bit about the global opinion (but we don't do the latter that rigorously).

Please submit a) your code (notebooks) *and* b) the results in a final output form (html or pdf).

You are welcome to answer some of the questions on paper but please include the result as an image in your final file. Note that you can easily include images in both notebooks and .rmd—besides of the code, both are just markdown documents.

Working together is fun and useful but you have to submit your own work. Discussing the solutions and problems with your classmates is all right but do not copy-paste their solution! Please list all your collaborators below:

1.

2. ...

World Values Survey

In this database we use World Values Survey data. The data is free to be downloaded from the webpage, just you have to sign up. It is a survey, conducted every few years in a number of countries. Here we use wave 6 data, mostly from 2013-2014. Note that not all countries are participating in each wave.

The questions revolve around different opinion topics, including trust, importance of democracy, work, religipn, family, gender equality, and nationalism. In this problem set we focus on what the respondents think about abortion: "Please tell if abortion can always be justified, never be justified, or something in between". The responses range between 1 – never justifiable, and 10 – always justifiable. Besides of the numeric range 1..10, a number of cases have negative codes (this applies to many variables). These are various types of missing information (-5: missing, -4: not asked, -3: not applicable, -2: no answer, -1: don't know). We treat all these as just missing below.

The version we use here is a little bit simplified, I have removed a large number of variables that are constructed from the other variables and hence highly collinear with the rest of the data.

I strongly recommend you to browse the documentation before you start, there are two large-ish documentation files provided.

1 Explore and prepare the data

As the first step, explore the data.

1. Load the data. How many responses and variables do we have?
2. Create a summary table over all responses for *V204*: is abortion justifiable. How many non-missing responses (i.e. positive answers) do you find? What is the opinion about the abortion among the global pool of respondents?
3. Now remove missings. We do it in two ways:
   (a) remove everything that are not positive integers for *V204* and *V2* (country).
   (b) for all other variables, remove the missings in the sense of missing value on computer. You may leave negative answers in the data, otherwise I am afraid your sample size collapses.
   What is the final number of observations?
4. In order to simplify the analysis below, create a new binary variable *abortion* as
$$\text{abortion} = \begin{cases} 1 & \text{V204} > 3 \\ 0 & \text{otherwise} \end{cases}$$
5. Compute (pearson) correlation table between *abortion* and all other variables in the data. There are many of these!
   Present these variables in descending order according to the absolute value of the correlation. It might look something like:

variable	correlation
abortion	1.000
x1	0.777
x2	-0.666
...	
x33	0.020
x44	-0.011

   Take a look at a few variables that have strong correlation with abortion. What do these represent?
6. convert country code *V2* into dummies. First rename *V2* to *country*. Thereafter use `pd.get_dummies` along these lines:

```
data2 = pd.get_dummies(data, columns = ['V2'])
```

   Afterwards, remove *country* variable from the data. How many rows/columns do you have now? How many country dummies does the data contain?
   Note that `get_dummies` creates a dummy for every category, so you have to remove one of these dummies in order to avoid perfect multicollinearity.

2 Find the best model

In this section your task is to find which model: k-NN, logistic regression, or SVM works best.

k-NN and SVM are sensitive to the distance metric, so you may also try to normalized versus non-normalized features. Check out `sklearn.preprocessing.normalize`. Logistic regression is agnostic with respect to the metric, but may benefit from more similar variable values for numerical reasons.

2.1 k-NN

First, use k-NN and experiment with a few different k-s.

1. Separate your training data into X (features), and **y** (target). Target will be the *abortion* variable, X are all the other features.

2. Split your training data into training/validation subsets. I recommend to use `sklearn.train_test_split`.

   This is high dimensional data and hence slow to fit, so you may want to start with a small training set. For instance, take both training and testing data to be 1000 observations only. Later slowly Increase the sample size till you feel it is getting too slow to work with. If model fitting is much slower than prediction (this is *not* the case for k-NN), you can easily go for validation data that is much larger than your training data.

   Note that you can give absolute dataset sizes to `train_test_split`. Several models (but unfortunately not SVM) in *sklearn* can also run in parallel, check out the `n_jobs` argument.

3. pick a k and fit the k-NN model. In sklearn you use something like

   ```
   from sklearn.neighbors import KNeighborsClassifier
   m = KNeighborsClassifier(5, n_jobs=-1)
   m.fit(X, y)
   ```

   where `X` are the features and `y` is the target, the abortion variable you constructed earlier.

4. Now predict the outcome on your validation set.

5. Present the confusion matrix (you may use `sklearn.metrics.confusion_matrix`). But write yourself a piece of code that computes accuracy, precision, recall, and F-score. Note: compute these values yourself, don't use any pre-packaged functions. You are welcome to compare your version with such as sklearn's `score`, but you have to show your own computations.

   Note: all these features include certain means of logical values, such as whether the prediction is correct. Sometimes `np.mean` is an almost trivial way to compute such things instead of `np.sum(x)/np.len(x)`, or even worse, counting values in a loop.

6. Adjust your sample size so that the process works reasonably fast (say, 1 min).

2.2 Logistic Regression

1. Now repeat the process above with logistic regression. As we have a myriad of features anyway, we are not going to do any feature engineering. Just a plain logistic regression.

   Note that as the optimization here is very different from the k-NN, you may want to use a different sample size.

2.3 SVM

Now repeat the process with support vector machines while choosing between a few different kernels and kernel options, such as *degree* for polynomial kernels.

I have mixed experience with *sklearn* version of SVM. I recommend to limit the number of iterations, initially maybe to just 1000, in order to ensure your model actually terminates.

1. pick a kernel and repeat the process above.

   Note that some kernels are slower than others, so be careful.

2. If your models worked like mine, you may have noticed that while accuracy seems all right, precision and recall are rather low. Explain what does such a phenomenon mean.

3 ROC curve: which estimator is the best

Now it is the turn for a make-it-yourself ROC curve. Consult James et al. (2015, p 148) before you start. In order to do this you have to predict not the target class (1/0) but the probability for the class being 1. Check out the `predict_proba` functions at *sklearn* for predicting these probabilities.

1. select the best model from each class of models you analyzed above (well, in case of logistic you probably just did a single run). If different indicators do not agree, you may follow what accuracy shows.
2. Pick a number of thresholds between 0 and 1 (for instance, 0, 0.1, 0.2, ...).
3. Predict the probability of the observation belonging to class 1. Use your validation data for this.
4. For each model, and for each threshold, treat your predictions to be 1 if the predicted probability is larger than the threshold.
5. Based on these predictions, compute false positive rate and true positive rate for each threshold
6. Plot these rates for each threshold and each model.
7. Comment your results. Which model is the best based on the ROC curve? Is there any clear winner?

4 How large a role does country play?

Here we switch from machine learning to social sciences. Public opinion differs from country to country, but also inside of countries. Does the fact that we include country code in data help us to substantially improve the predictions?

You pick the best ML method according to the ROC curve above. You estimate two sets of models: one with country information included, and one where it is removed. Is the former noticeably better than the latter?

1. pick your best ML method based on the ROC curve. Predict the abortion variable using all the features, including country dummies and report the accuracy. Essentially you repeat here what you did above.
2. Now remove all the country dummies, but keep the other variables intact. And repeat.

   Note: keep the training/testing data the same as in the previous step: we are interested in the effect of country, not the effect of training data size here.
3. Comment what you found. Does country information help to improve the prediction substantially?

Well... we are perhaps a bit unfair here. Obviously we can quite precisely predict someone's views on abortion if we know the person's views on a few closely related topics.

1. Remove the few highly correlated variables: *V203A*, *V207*, *V206*, *V203*, *V205*, revolving around the views on homosexuality, prostitution, divorce, and such.

   Obviously, there may be many more variables that are either good proxies for views of abortion, or highly correlated with the country (say, certain questions were not asked in certain countries and have all the corresponding missing code). So our estimates should still be treated as an exercise.

2. ...and repeat what you did above.

3. Does the country effect change?

References

James, G., Witten, D., Hastie, T., Tibshirani, R., 2015. An Introduction to Statistical Learning with Applications in R. Springer.